Distributed Planning in Constellations of Autonomous Spacecraft

Dr. T. J. Grant and Dr. A. H. Broeils
Origin Nederland B.V., Technical Automation/Command and Control,
P.O. Box 1444, 3430 BK Nieuwegein, The Netherlands
Tel: +31(0)30 608 88 88
Fax: +31(0)30 606 05 77
E-mail: {Tim.Grant, Adrick.Broeils} @nl.origin-it.com
Url: http://www.origin-it.com

1. ABSTRACT

This paper describes the application of Origin's generic simulation architecture and complementary toolset to a demonstrator facility for planning in satellite constellations. The architecture was designed to support widespread applicability of real-time simulation technology on low-cost platforms like PCs with Windows NT. The goal of the facility is to simulate and analyse planning problems that will occur in constellations of autonomous satellites. We present an innovative concept for mission planning based on a distributed planning algorithm. We find that Origin's generic simulation architecture is well suited for building a distributed planning tool.

2. INTRODUCTION

The purpose of this paper is to describe an innovative concept and implementation for mission planning and scheduling in constellations of autonomous spacecraft. The need for an innovative concept follows from the observed market trends that spacecraft will become smaller, highly autonomous, and will often operate in constellations, gathering and returning massive volumes of data to Earth. We foresee tuning problems both for downlinking data of several spacecraft down to one ground station as for uplinking detailed schedules (in the form of Tele-Command (TC)-sequences) to all satellites in a constellation. Additionally, the planning effort for constellations with tens or hundreds of satellites will be large.

Section 3 provides more information on the background of the tuning problems for satellite constellations. In Section 4 we describe various concepts for mission planning and argue that decentralised planning is the most promising concept to solve these problems. Under this concept, a plan is negotiated by the satellites in the constellation, rather than detailed in the Mission Control Centre (MCC). A possible negotiation protocol is introduced. Section 5 illustrates how Origin's generic simulation architecture is used to build an analysis tool for planning in satellite constellations. Finally, in Section 6 we draw conclusions.

3. PLANNING IN SATELLITE CONSTELLATIONS

The current generation of (remote-sensing) satellites is designed around individual, large satellites carrying many instruments. Space exploration and applications in the 21^{st} century will be significantly different from what we see today.

To save development time and cost, increasing use will be made of small satellites, typically equipped with only one or two instruments. In case of failure these *smallsats* can be replaced for a much lower price than the present day large satellites. Smallsats will have relatively cheap designs, and can be "mass" produced. The low cost of individual smallsats allows the use of spare satellites as part of the mission plan (as is the case in present-day telecommunication constellations systems, like Iridium [1]).

Smallsat will be able to operate on an individual basis, but a considerable portion will also co-operate in constellations, formations and networks to fulfil certain missions. Constellations will provide the coverage needed in many missions, like the Global Positioning System (GPS) for navigation. Satellite networks are already used in telecommunication, like the Iridium system. Formations consist of (generally) non-identical, but complementary spacecraft and together provide new products. For instance, satellite formation can be used to observe simultaneously the same target under different geometry's, with various instruments. In this paper we will use the word *constellations* for constellations, networks and formations of satellites.

Smallsats will become increasingly more autonomous. Autonomy will decrease the operating cost, since satellites will operate more independently from ground control.

Presently, nearly all planning for individual satellites and satellite constellations is done centrally in the

Proc. Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space, 1-3 June 1999 (ESA SP-440)

MCC. In the MCC the end-user goal is decomposed into sub-goals and allocated to the appropriate smallsats. Each smallsat's subgoal-set is decomposed into TC-sequences and uplinked by a ground station to the appropriate smallsat and executed.

This approach will become increasingly difficult for large constellations of satellites, because of computational complexity. Conflict resolution, scheduling and uplinking will become a bottleneck. Especially in cases where a number of satellites have to co-operate to perform a certain task, the planning and detailed control of the individual satellites from the ground becomes nearly impossible. Eventually, the mission planning process will no longer be able to react sufficiently swiftly to new user requests and to unforeseen situations happening in the constellation.

In addition, new resource bottlenecks arise in constellations. Spacecraft will compete in using the limited number of ground stations to download their data. The situation will be aggravated for smallsats, which only may carry one or two payloads. Besides, smallsats may have to cooperate in groups for limited period of time to perform some operations.

Therefore, we foresee two tuning problems:
1) planning and controlling the individual satellites for co-ordinated actions from the ground will be nearly impossible, and 2) downlinking of data by multiple satellites after a co-ordinated action to a single ground station will lead to data congestion. Clearly, the existing off-line, ground-based planning facilities will be unable to cope with generating fully-detailed TC-sequences for constellations containing up to tens or even as many as a few hundred satellites (as proposed in the Teledesic system[2]).

4. SOLUTIONS

4.1 Planning concepts

Origin initiated a project to investigate the tuning problems mentioned in the previous section. The end-goal of the project is to develop of a planning facility for satellite constellations. The planning facility can analyse planning concepts by translating a high-level user request into a detailed planning of required actions for the individual satellites in the constellation.

Planning can be done in various ways and at different locations. To create a plan the following approaches can be used:

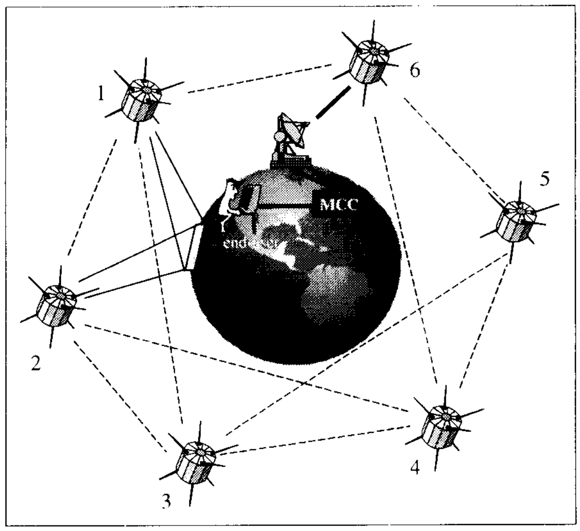

Figure 1 Decentralised planning approach

- *Centralised planning*: In the traditional approach a planning for individual satellites and satellite constellations is done centrally in the MCC. In the MCC the end-user goal is decomposed into sub-goals and allocated to the appropriate satellite. Each satellite's subgoal-set is decomposed into TC-sequences and uplinked by a ground station to the appropriate satellite, and executed.
- *Decentralised planning*: In a decentralised planning scenario the MCC translates the end-user request into a high-level command (HLC) that is uplinked to one of the satellites in the constellation. The satellite decomposes the high-level command into sub-goals and distributes these among the other satellites (through Inter-Satellite Links or ISLs) in order to achieve an internal planning of the required actions. Once the sub-goals are allocated to specific satellites, every satellite will generate its own TC-sequences and will execute them. Decentralised planning seems to be the most promising approach, given the increase in the number of small autonomous satellites foreseen in the future.

In addition there is an intermediate approach between these two extremes:

- *Planning by distributed simulation*: Planning is done in the MCC through distributed simulation of the constellation. The MCC sends a user request in the form of a HLC to the simulated constellation. The simulated constellation negotiates a plan, detailing this fully as TC-sequences. The TC-sequences are extracted from the distributed simulation and uplinked to the appropriate satellites in the real constellation.

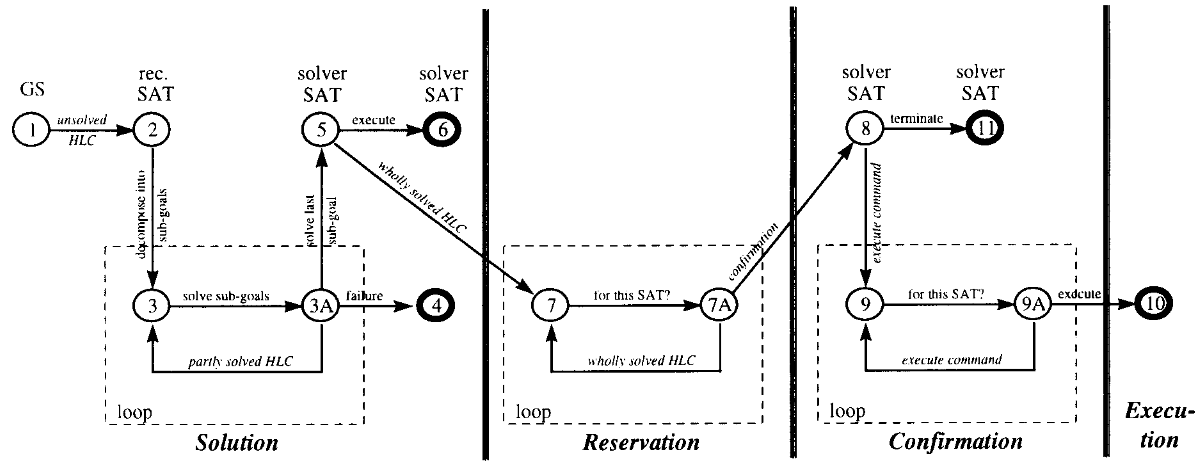

Figure 2 Negotiation steps

We argue that decentralised planning is the most promising solution for the tuning problems mentioned above. To show that decentralised planning is feasible we use distributed simulation with satellites modelled as autonomous agents as the foundation for the planning facility. With this approach, two separate solutions are still possible, depending on the capabilities of the satellites in the constellation. If the real satellites are capable to communicate with each other, the planning facility could serve as a validation facility which validates the feasibility of the request faster than real-time. If the real satellites do not have ISLs, the detailed planning produced by the facility can be send to the satellites in the traditional approach. At this stage of the project we see the planning tool as a useful tool for mission engineers and specialist to be used for feasibility studies.

A schematic picture of the mentioned decentralised approach is shown in Figure 1. Here the high-level request is (for example) a request for an observation of a specific target on Earth at a specific time and using two specific instrument. The ground station uplinks the HLC to the first satellite that comes within reach (nr. 6). The satellites in the constellation start negotiating until it is determined that satellites 1 and 2 have the required instruments and are above the target at the specified time. If satellites 1 and 2 accept the command (i.e. allocate time and resources in their timelines), they decompose their portion of the HLC into TC-sequences and perform the observations.

The appropriate technology to implement a decentralised planning scenario exists in Distributed Artificial Intelligence (DAI), where it is known as Multi-Agent Systems (MAS). For our facility it means that each satellite, MCC, and ground station can be modelled as an agent, with the satellites having intelligent planning and negotiation capabilities. Since we chose to build a facility for a remote-sensing mission, we also need to simulate the observable targets.

4.2 Negotiation and planning process

In the decentralised planning scenario, the planning is done through negotiating between satellites in the constellation. The negotiating process goes through a number of steps[1], each step involving a different set of possible messages. Consequently, several processes must occur in a satellite, depending on the type of message it receives.

We see a HLC as a combination of several sub-commands or sub-goals that need to be "solved".The first satellite that receives the HLC (uplinked from the ground) will split it into several sub-goals. We assume for simplicity that a satellite can solve a sub-goal independently from any other satellite. During the planning process the complete HLC command is passed on from one satellite to another, with solutions to the sub-goals being progressively added by each satellite.

We identify four steps in the negotiation process:

- *Solution.* The HLC is decomposed into sub-goals and actions by the satellite that initially receives it. The input is an unsolved HLC. After decomposition, the set of sub-goals is broadcast to the other satellites in the constellation for solution. One or more satellites co-operate in proposing solutions to the sub-goals until the complete set is solved or failure. Output is a wholly solved HLC or a message informing the MCC of failure. If the output is a wholly-solved HLC, then the satellite achieving the solution becomes the "Solution Master" (we will call it the solverSAT).

---

[1] Like, for example, Tender, Bid, and Award in the Contract Net Protocol [3].

- *Reservation.* The co-operating satellites assign the actions by making resource reservations in their timelines. The inputs are one or more wholly solved HLCs, the outputs are confirmations to the solverSAT.
- *Confirmation.* The solverSAT determines which potential solution will be adopted. As there will be no optimisation in this project, the first potential solution obtained will be adopted. The input is a confirmation message and the output is an execution message to the satellites involved.
- *Execution.* The satellites involved in the confirmed solution execute the actions using the resources reserved in their timelines. The input is an "execute" command, and the output is a message to the MCC to report execution.

The four major steps in the negotiation process are schematically depicted in Figure 2. The numbers model states in the negotiation process, and the arrows indicate transitions between these states. The uplink of the unsolved HLC by the ground station is the starting point of the whole process (State 1). End-states are indicated by the "fat" circles (states 4, 6, 10, 11). Roughly half of the transitions correspond with message exchange *between* satellites (indicated by italic type-face in Figure 2), the rest are transitions *within* a satellite. States in the top row of the diagram occur in objects that (temporarily) have a special function in the constellation, i.e. the satellite that receives the unsolved HLC from the ground station, the solverSAT (the satellite that solves the last sub-goal), and the ground station. States in the bottom row occur in some or all of the other satellites in the constellation. There are three loops in the negotiation process. In these loops messages are spread through the constellation until all sub-goals are solved (in "solution" loop) or until message reaches the correct satellite (in "reservation" and "confirmation" loops).

5. TECHNOLOGY

5.1 Architecture

Origin has defined a generic architecture to support a wide spread applicability of low-cost real-time simulation technology. The architecture (described in [4]) is designed to support re-usability, extendibility, and scalability. Two types of simulations are recognised, the virtual world (consisting of a number of autonomous virtual objects) and the simulator (consisting of several components that have detailed knowledge of each other).

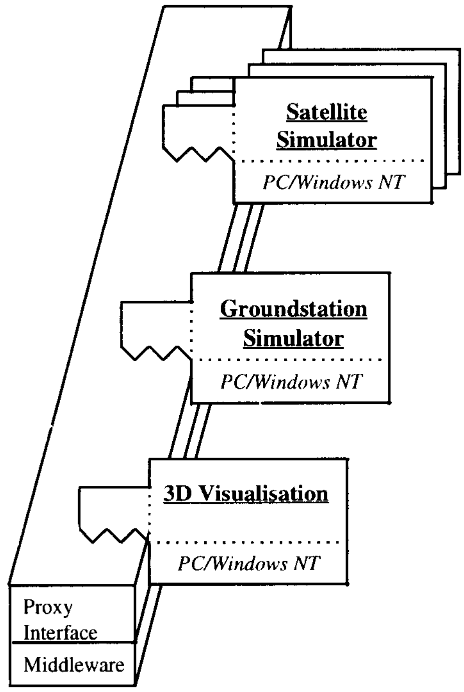

Figure 3 Virtual world simulation architecture

For the constellation planning simulations the virtual world part of the generic architecture is used (Figure 3). It contains a middleware layer based on a proxy design pattern. The proxy interface simplifies interfacing with the simulated world, by automatic creation of representatives of all other objects within the context of an application. This middleware approach abstracts the application from the communication infrastructure as it maps the simulated world directly into the context of the application. At the virtual world three categories of applications are found, all connected to a proxy interface layer:

- *Simulator Applications* are applications that simulate the behaviour of an object within the virtual world, regularly updating their state, monitoring the state of other objects and interacting with other objects. In the planning facility satellites, ground stations, MCC, targets, and even the Earth are examples of simulator applications. Their internal construction are hidden from other objects, but objects publish information when interacting with other objects.
- *Command & Control (C^2) Applications* are applications that do not represent an object in the simulated world but can interact with objects. A typical example is a telecommand and telemetry station, which receives information from a satellite and can uplink new commands, but is not an object with a representation in the simulated world. In the planning facility the C^2 and simulator applications are actually combined in the MCC. It receives a request from the user, and translates this to a HLC for the constellation.

- *Generic Applications* are tools that can be used within every distributed simulation. Typically these respond to control and monitoring interfaces. An example is a 3D visualisation tool, which can visualise the state and interactions of objects within the virtual world.

For these type of simulations the coherence is low, and a variable number of objects is supported which can even be varied during a simulation. This last feature is useful for a satellite constellation simulation if the need arises to evaluate plans in case of launching new satellites or error situations.

5.2 Implementation

The virtual world part of the architecture is traditionally called a distributed simulation architecture. For this part we use the High-Level Architecture (HLA) as the middleware layer. HLA was defined by the US Department of Defense (DoD). The DoD declared that HLA was to be the standard technical architecture for all DoD simulations. As such it is anticipated that HLA will also quickly become a standard architecture in the space industry. HLA describes a standard interface for simulators, which allows them to interact over a network, using a global notion of *time*, i.e. a global ordering is given to all events in the simulation.

During a simulation each "physical" object (e.g. satellite, ground station, planet etc.) publishes its position, velocity and other information through the HLA services to all other simulators (or federates in HLA terminology) in the federation (A federation in our case is a collection of all federates in the simulation) at regular time intervals. This information can be used by a federate, for example, to see if communication between two federates is possible.

Origin developed the proxy interface ADS on top of the HLA middleware layer. It creates a representative of each instance of the classes derived from the "physical" objects" class within the local environment of each application (e.g. satellite, visualisation tool etc.). This is illustrated in Figure 4. Each simulator obtains status updates (position, orientation) of other objects through the proxy interface.

The planning processes as described in Section 4.2 are implemented as member functions of the physical object classes. Message exchange between satellites and between satellites and ground stations are implemented as HLA interactions and exchanges of object ownership. Interactions are short-duration events that can be discovered by all other simulators.

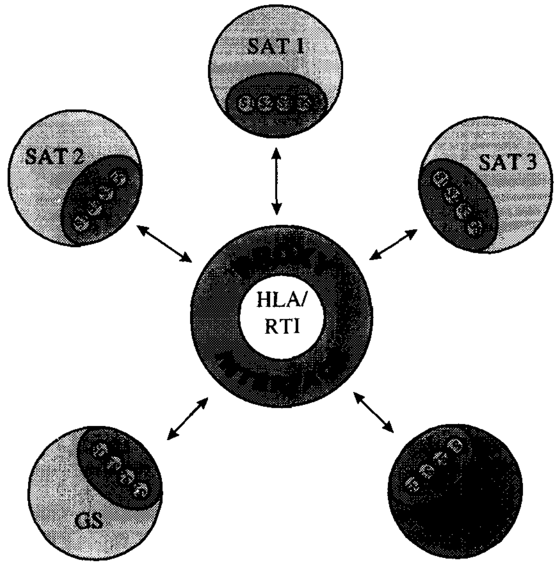

Figure 4 ADS Proxy Interface to HLA services

5.3 AVE: A 3D stealth application

For most distributed simulation, a general 3D view of the virtual world is required, which allows inspection of the virtual world without influencing it. Such a tool is often referred to as a 'stealth'.

Based on the object-oriented proxy interface and an onward development in creating an object-oriented Virtual Environment, a low-cost 3D stealth (named AVE) has been created. The stealth supports the dynamic character of distributed simulations, where multiple instance of a type of object can exist. Each "physical" object automatically receives a connected camera, orientation axes, head-up display, trail facility, interaction display and message display facility. In our architecture (Figure 3 and 4), the VE tool is also a federate receiving the attribute updates and interaction of all other federates in the federation, enabling it to project satellites and message between satellites and other objects on the display. Figure 5 shows a screendump from the implemented Planning in Autonomous Constellations Tool (PACT).

6. CONCLUSIONS

With the growth of the number of autonomous smallsats operating in constellations we foresee severe planning problems if one tries to perform all planning actions centrally in the MCC. We propose a distributed planning concept where the planning is autonomously negotiated by the spacecraft in the constellation. We show that this concept can be analysed well using a distributed simulation facility that we implemented using Origin's generic architecture for low-cost simulations. Especially, the scalability issue is of importance here: to simulate a much larger system one can simply add a few low-cost PCs to the simulation environment.

7. REFERENCES

1. See http://www.iridium.com
2. See http://www.teledesic.com
3. R.G. Smith, R. Davis, *Proc. 1979 Int. Conf Cybern Soc*, Oct. 1979
4. J. Keijzer, L. Bremer, R. Trompert, *A Satellite Constellation Planning Tool: Application of a low-cost real-time simulation architecture*, Proceedings, 1999 European Simulation Multiconference, Warsaw, Poland, 1-3 June 1999.

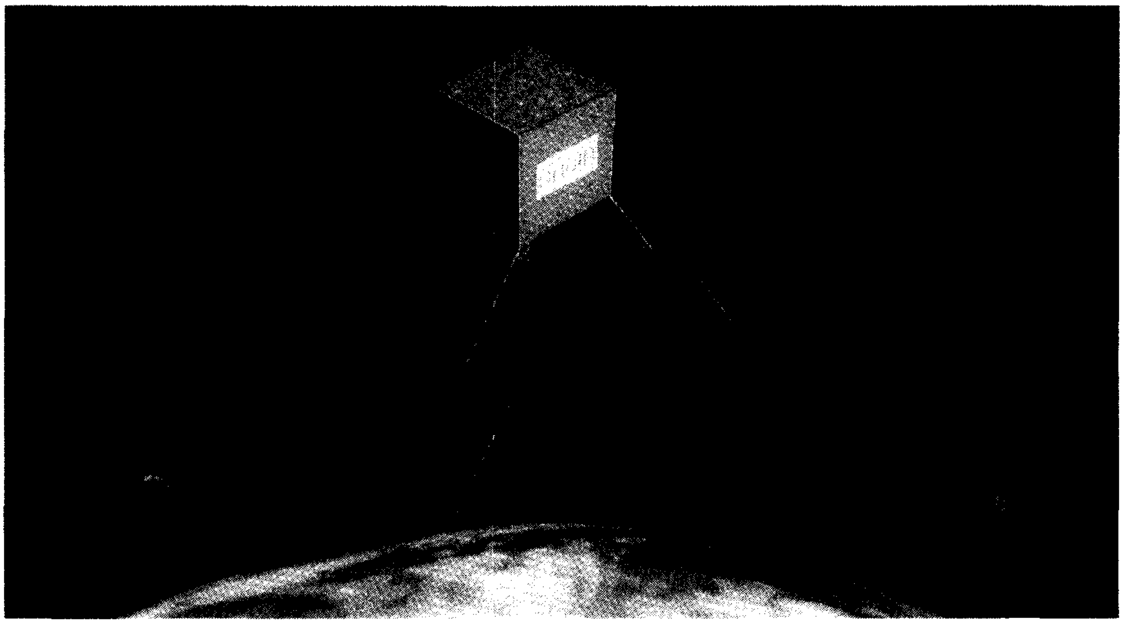

Figure 5. Screendump from Planning in Autonomous Constellations Tool.